FVPP: Vremenska logika i računanje skupova stanja

Linearna vremenska logika (LTL)

Potpuna logika vremenskog grananja (CTL*)

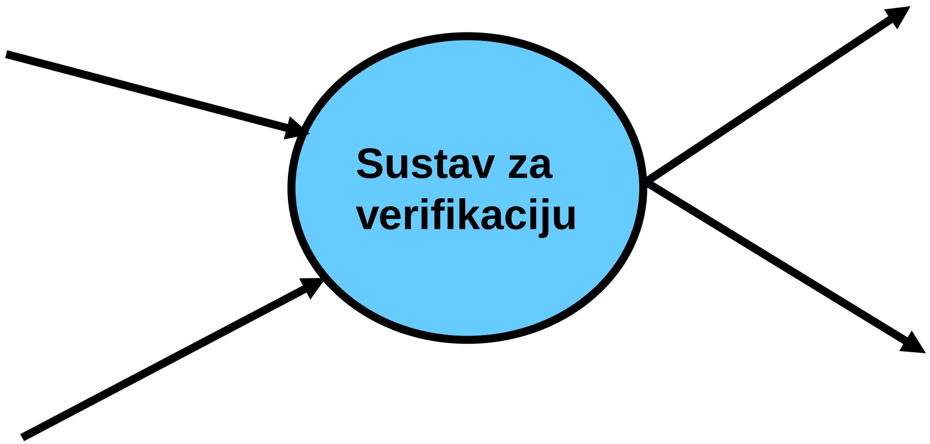
Eksplicitni postupci računanja skupova stanja koja zadovoljavaju CTL specifikaciju

Pripremio: izv. prof. dr. sc. Alan Jović

Ak. god. 2022./2023.

Provjera modela (engl. *model checking*)

***I* = Implementacija (model sustava koji se verificira). Izraženo povezanim strojevima s konačnim brojem stanja (FSM).**



DA = model sustava logički zadovoljava specifikaciju

NE (+ ispis traga (engl. *trace*) pogrešnog izvođenja programa)

***S* = Specifikacija (željeno ponašanje). Izraženo najčešće u vremenskoj logici - CTL ili LTL**

Simbolički opisujemo:

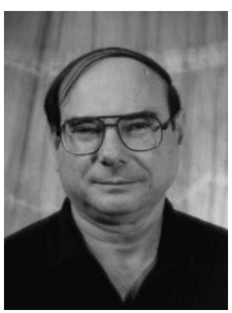
$$\boxed{I \models S}$$

FVPP: Propozicijska linearna vremenska logika

Engl. *Propositional Linear Temporal Logic*

PLTL (najčešće samo LTL)

A. Pnueli: "The temporal logic of programs" (1977.)



Pretpostavka primjene (vrijede i za CTL)

- Promatramo sustave koji se mogu modelirati strojevima s **konačnim brojem stanja**

- Promatramo **reaktivne programe (neterminirajuće)** – kontinuirano reagiraju na okolinu (operacijski sustavi, sustavi upravljanja procesima i sl.)

- Analiza ponašanja duž **potencijalno beskonačnih putova izvođenja**.

Prisjetimo se...

- Kontekst vremenske logike CTL:
- Kripkeova struktura (model M) promatra se kao beskonačno stablo počevši od početnog stanja s0 (“stablo se odmota”).
- U svakom stanju moguć je prijelaz u jedno od **mogućih više stanja**.
- Eksplicitno se kvantificiraju svi putovi izvođenja (**A**) ili barem jedan put izvođenja (**E**).
- Formula vremenske logike CTL je istinita ako je istinita barem za jedan put izvođenja (**E**) ili za sve putove izvođenja (**A**), ovisno o vrsti specifikacije.

Kontekst vremenske logike LTL

- Kripkeova struktura (M) se dekomponira u **pojedinačne beskonačne sekvence** – putove izvođenja programa.

- Ponašanje sustava je **kolekcija beskonačnih sekvenci prijelaza** (engl. *infinite transition sequences*)

- Svako stanje u pojedinoj sekvenci ima **samo jednog sljedbenika**.

  sekvenca 1: $s0 \to s1 \to s2 \to s3 \to s4 \to s5 \to s6 \to s7 \to \ldots$

  sekvenca 2: $s0 \to s1 \to s2 \to s7 \to s4 \to s5 \to s6 \to s2 \to \ldots$

  sekvenca *n*: $\ldots$

- LTL formula je istinita za neki sustav (Kripkeovu strukturu) ako vrijedi za sve pojedinačne putove izvođenja (za sve sekvence) toga sustava. Kvantifikator "A" je implicitan. Nema kvantifikatora "E".

Linearna vremenska struktura

- Jedna beskonačna sekvenca s početnim stanjem i označavanjem propozicijskih simbola koji vrijede u pojedinim stanjima naziva se **linearna vremenska struktura**.

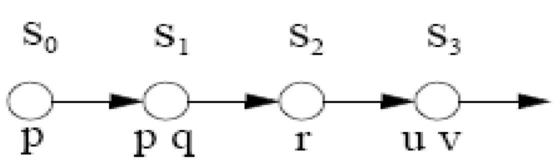
- Formule logike LTL se interpretiraju po svim beskonačnim sekvencama (linearnim vremenskim strukturama).

- AP: skup atomičkih propozicijskih simbola
- **Linearna vremenska struktura** dana je trojkom π **= (S, x, L)**

S:	konačan skup stanja
x: $\boldsymbol{N} \rightarrow \mathbf{S}$	beskonačna sekv. prijelaza = vremenska crta ($x = s_0, s_1, \ldots$)
L: $\mathbf{S} \rightarrow 2^{\mathbf{AP}}$	označavanje stanja skupom propozicijskih simbola.

Primjer:

x:



$$L(s_0) = \{p\},\ L(s_1) = \{p, q\},\ L(s_2) = \{r\},\ L(s_3) = \{u, v\},\ldots\ldots$$

Vremenski operatori u LTL-u

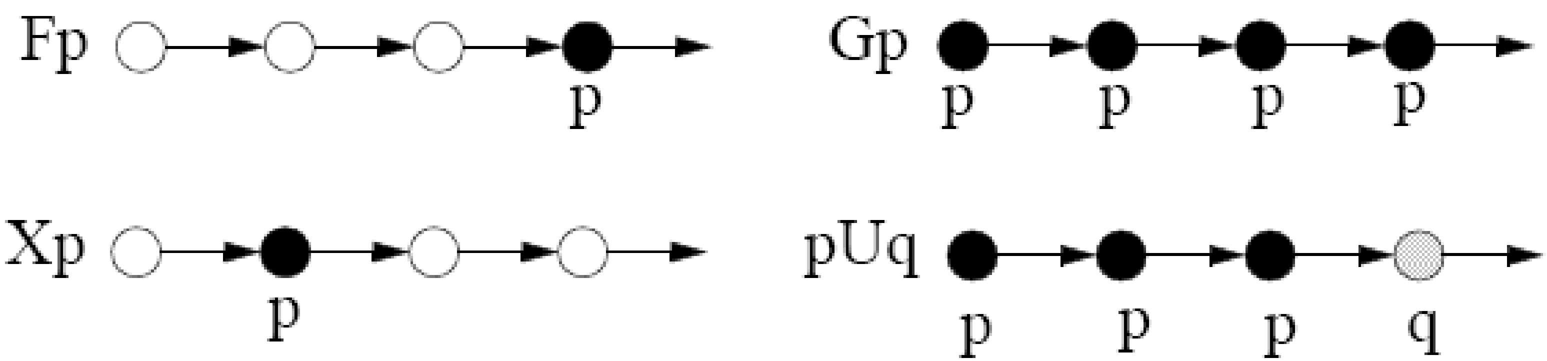
Uporaba vremenskih operatora (bez kvantifikatora):

F*p* (“eventually *p*”, “finally *p*”) - konačno p

G*p* (“always *p*”, “henceforth *p*”) - uvijek p

X*p* (“next time *p*”) - u sljedećem koraku p

*p***U***q* (“*p* until *q*”) - p dok ne počne vrijediti q



Provjera modela: implicitno provjeravamo **sve** putove u Kripkeovoj strukturi (kvantifikator A je implicitan).

Formalna sintaksa (P)LTL-a (1/2)

(1) Atomičke propozicije su formule.
(2) Ako su p i q formule, tada su $p \wedge q$, $\neg p$, $p \cup q$, Xp formule.

Ostale formule mogu se izvesti:

$p \vee q$	ekvivalentan oblik	$\neg(\neg p \wedge \neg q)$,
$p \Rightarrow q$	ekvivalentan oblik	$\neg p \vee q$,
$p \equiv q$	ekvivalentan oblik	$(p \Rightarrow q) \wedge (q \Rightarrow p)$,
true	ekvivalentan oblik	$p \vee \neg p$,
false	ekvivalentan oblik	$\neg$*true*,
Fp	ekvivalentan oblik	(*true* $\cup$ p),
Gp	ekvivalentan oblik	$\neg F \neg p$.

LTL dozvoljava Booleove kombinacije i ugniježđivanje vremenskih operatora

Formalna sintaksa LTL-a (2/2)

Prioritet unarnih i binarnih operatora:

1. Unarni operatori ($\neg$, X, F, G) povezuju najčvršće
2. U - binarni operator
3. $\wedge$ - konjunkcija
4. $\vee$ - disjunkcija
5. $\Rightarrow$ - implikacija

Primjer suvišnih zagrada:

$(F\ (p \Rightarrow (G\ r)) \vee ((\neg q)\ U\ p)) \equiv F\ (p \Rightarrow G\ r) \vee \neg q\ U\ p$

Preporuka: **upotrebljavaj zagrade zbog jasnijeg razumijevanja**, iako mogu biti suvišne.

Formalna semantika (P)LTL-a

Formula φ dana u vremenskoj logici LTL ima značenje u odnosu na **svaku pojedinu linearnu vremensku strukturu** π **= (S, x, L)**.

π**, x** $\models \varphi$ - u strukturi π formula φ je istinita za vremensku crtu **x**.
Crta (put) **x** logički zadovoljava ($\models$) formulu φ.

Neka je:

x - put (vremenska crta) koja započinje u s_0 , **x** = $s_0, s_1, \ldots$

$\mathbf{x_i}$ - put (vremenska crta) koja započinje u s_i , $\mathbf{x_i} = s_i, s_{i+1}, \ldots$

Tada:

$\pi, x \models a$	akko	$a \in L(s_0)$, AP **a je istinit u** $\mathbf{s_0}$.
$\pi, x_i \models b$	akko	$b \in L(s_i)$, AP **b je istinit u** $\mathbf{s_i}$.
$\pi, x \models \varphi \wedge \psi$	akko	$\pi, x \models \varphi$ i $\pi, x \models \psi$
$\pi, x \models \neg\varphi$	akko put **x** u strukturi π ne zadovoljava φ	
$\pi, x \models \varphi \, \mathbf{U} \, \psi$	akko	$\exists_i \, (x_i \models \psi)$ i $\forall_{k<i} \, (x_k \models \varphi)$
$\pi, x \models \mathbf{X}\varphi$	akko	$x_1 \models \varphi$ u sljedećem stanju je φ **= T**
$\pi, x \models \mathbf{F}\varphi$	akko	$\exists_i \, (x_i \models \varphi)$ postoji neko stanje gdje je φ **= T**
$\pi, x \models \mathbf{G}\varphi$	akko	$\forall_i \, (x_i \models \varphi)$ u svakom stanju φ **=T**

Konvencija: **sadašnje stanje je uključeno u buduće stanje.**

Načini iskazivanja beskonačnosti u LTL-u (engl. *LTL infinitary modalities*)

p: formula u LTL-u

$F^{\infty} p \equiv GF\ p$ ("globally finally *p*", "infinitely often *p*")

$s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow s_4 \rightarrow s_5 \rightarrow s_6 \rightarrow s_7 \rightarrow \ldots$

$\neg p \quad \neg p \quad p \quad \neg p \quad p \quad \neg p \quad p \quad \neg p$

"infinitely often *p*" – "**beskonačno često se pojavljuje p**"

$G^{\infty} p \equiv FG\ p$ ("finally globally *p*", "almost everywhere *p*")

$s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow s_4 \rightarrow s_5 \rightarrow s_6 \rightarrow s_7 \rightarrow \ldots$

$\neg p \quad \neg p \quad \neg p \quad \neg p \quad p \quad p \quad p \quad p$

finite ¬p **infinite p**

"Konačno globalno se pojavljuje p": nakon konačnog broja stanja u kojime p ne mora biti istinit, slijedi beskonačan niz stanja u kojima je p=TRUE.

Ispravnost LTL-formula

Primjeri sintaktički ispravnih LTL-formula:

$p \Rightarrow \mathrm{F}q$

$\mathrm{G}\,(p \Rightarrow \mathrm{F}q)$

p U (q U r)

XXG p

$[\, p \wedge \mathrm{G}\,(p \Rightarrow \mathrm{X}p)\,] \;\Rightarrow\; \mathrm{G}\,p$

Primjeri sintaktički neispravnih LTL-formula:

$\mathrm{U}\,r$ (U nije unarni operator)

$p\,\mathrm{G}\,q$ (G nije binarni operator)

$\mathrm{EG}\,p$ (E kao kvantifikator ne postoji u logici LTL)

$\mathrm{A}\,p\,\mathrm{U}\,q$ (A se ne navodi eksplicitno kao kvantifikator u logici LTL)

Značajne valjanosti u LTL-u

Vrijede za sve linearne vremenske strukture π **= (S, x, L)**

Neka su p, q formule u LTL-u.

$\models G \neg p \equiv \neg F p$

$\models F \neg p \equiv \neg G p$

$\models X \neg p \equiv \neg X p$

$\models F^{\infty} \neg p \equiv \neg G^{\infty} p$

$\models G^{\infty} \neg p \equiv \neg F^{\infty} p$

GF = $F^{\infty} p$ – "beskonačno često p

FG = G^{∞} – "konačno (nakon nekog vremena) globalno (stalno) p"

$\models p \Rightarrow F p$

$\models G p \Rightarrow p$

$\models X p \Rightarrow F p$

$\models G p \Rightarrow X p$

$\models G p \Rightarrow F p$

$\models G p \Rightarrow XG p$

$\models p \, U \, q \Rightarrow F q$

$\models G^{\infty} q \Rightarrow F^{\infty} q$

(zadnja izjava: ako nakon nekog stanja globalno q, tada i beskonačno često q)

Distributivnost i rekurzije u LTL-u

Distribucija preko logičkih vezica:

$\models \quad G\,(p \land q) \quad \equiv \quad (G\,p \;\land\; G\,q)$

$\models \quad (p \land q)\;U\;r \quad \equiv \quad ((p\;U\;r) \land (q\;U\;r))$

analogno za X, F

Primijetiti da u CTL-u takva distribucija nije dozvoljena:

CTL: $A[(p \land q)\;U\;r\,] \neq A[(p\;U\;r) \land (q\;U\;r)\,]$

Rekurzivni izrazi:

$\models \quad F\,p \quad \equiv \quad p \;\lor\; XF\,p$

$\models \quad G\,p \quad \equiv \quad p \land XG\,p$

$\models \quad (p\;U\;q) \;\equiv\; q \lor (p \land X\,(p\;U\;q))$

Preslikavanje rečenica prirodnog jezika u LTL (1/2)

1. Nije moguće doći u stanje gdje je *start* istinito a *spreman* nije istinito.

   $\mathrm{G}\neg(start \land \neg spreman)$

2. Za svako stanje vrijedi: ako se pojavi zahtjev, on će konačno biti *prihvaćen*.

   $\mathrm{G}(pojavio\_zahtjev \Rightarrow \mathrm{F}\ prihvaćen\_zahtjev)$

3. Neki proces je omogućen beskonačno često na svim putovima izvođenja.

   $\mathrm{GF}\ proces\_omogućen$

4. U svakom slučaju, određeni proces će permanentno biti zaustavljen.

   $\mathrm{FG}\ proces\_zaustavljen$

5. Iz svakog stanja sustava **moguće** je doći u stanje gdje vrijedi *reset*.

   To se ne može izreći LTL-logikom, jer LTL ne može izravno potvrditi postojanje (moguće) nekog određenog puta ili putova (to naravno može CTL kvantifikatorima puta A i E).

Preslikavanje rečenica prirodnog jezika u LTL (2/2)

6. Lift **može** ostati stajati na …

Ove sve izjave ne mogu se izreći LTL-logikom, jer LTL ne može izravno potvrditi postojanje (moguće, može) nekog određenog puta ili putova (to naravno može CTL kvantifikatorima puta A i E).

Problem se djelomično može riješiti **negacijom upita**:

- Provjera postoji li put iz *s* koji zadovoljava LTL-formulu φ svodi se na provjeru da li svi putovi zadovoljavaju $\neg\varphi$.
- Ako svi putovi zadovoljavaju $\neg\varphi$, onda ne postoji put koji zadovoljava φ.
- Međutim, provjera značajki sustava u kojem postoji mješavina egzistencijskih i univerzalnih kvantifikatora ne može se riješiti na gornji način (negacija opet daje mješavinu E i A).

Proširenja LTL-a

1. **LTL s konačnim linearnim vremenskim strukturama** (odmotavamo samo dio puta: prvih *k* stanja – koristi se za ograničenu provjeru modela (engl. *bounded model checking*).

2. Promjena semantike modaliteta:

   2.1. (p U q) je istinita sve dok vrijedi ($p \wedge \neg q$), tj.

   q može stalno biti neistinit, tj. ne mora biti istinit negdje u budućnosti ("weak until"): (p W q)

   2.2. (p U q) je istinita akko u budućem trenutku (**ne sada**)

   q je istinit (q mora biti istinit **striktno u budućnosti**).

3. Proširenje s logikom predikata prvoga reda (FOLTL)

4. Proširenje s prošlim vremenom (ptLTL) - "past time LTL" – ima istu izražajnu moć kao i LTL sa samo budućim vremenom

…

Usporedba CTL-a i LTL-a (1/3)

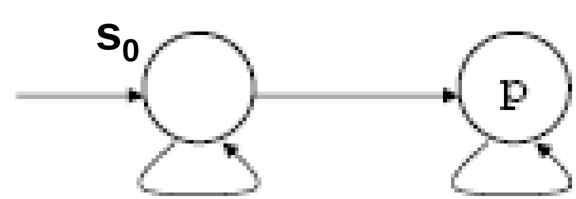
1. **Različita i neusporediva izražajnost** (ekspresivnost).
2. CTL **eksplicitno kvantificira putove**
3. LTL može selektirati **sve pojedinačne putove iz nekog stanja** koji zadovoljavaju LTL-formulu
4. Neke formule u CTL-u nije moguće izraziti u LTL-u i obratno.
5. U LTL-u **su složenije procedure provjere modela** ali jednostavnije neke druge procedure (npr. provjera valjanosti formula).

Usporedba CTL-a i LTL-a (2/3)

CTL-formula AG (EF *p*) = “iz kojeg god stanja da krenemo možemo doći (postoji barem jedan put) do stanja gdje je p istinit” **nema ekvivalenta u LTL-u**.

Primjer: Neka je φ LTL-formula takva da je A[φ] navodno ekvivalentno AG(EF *p*) , tj. za sve putove.

Za model kao na slici:



- φ kao LTL-formula GF p ne vrijedi za put (beskonačnu petlju) stalno u s_0 (izdvojen put) jer mora vrijediti za sve pojedinačne putove.
- φ = AG (EF *p*) kao CTL-formula vrijedi jer EF *p* vrijedi za oba stanja.

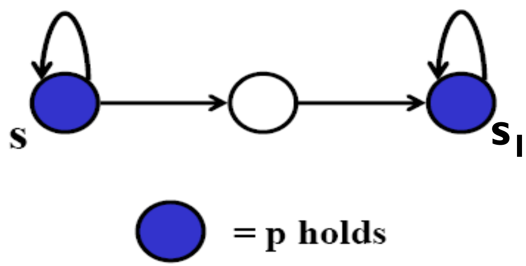
Usporedba CTL-a i LTL-a (3/3)

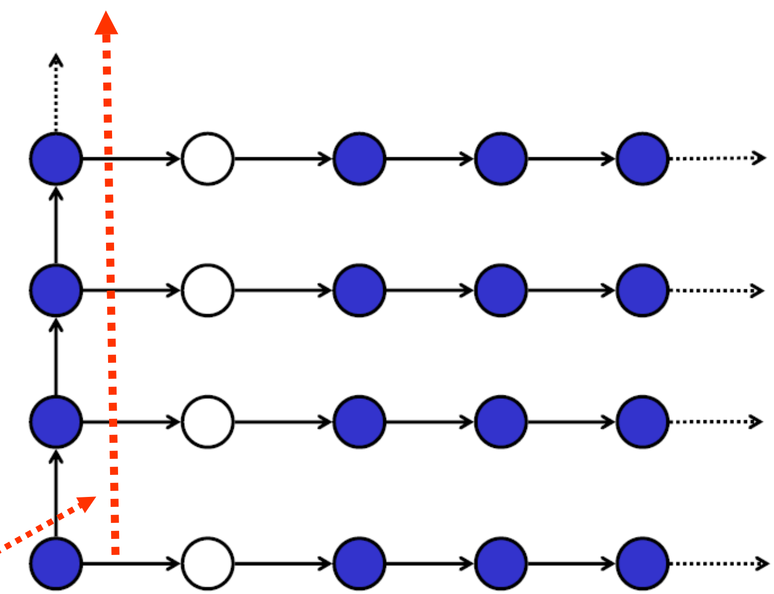
LTL-formula FG p nije ekvivalentna CTL-formuli AF (AG p)

FG p = konačno (nakon konačnog broja koraka) globalno (stalno) p

AF (AG p) = na svim putovima uvijek dolazimo konačno do stanja iz kojega je dalje stalno p=T.

U prikazanom modelu FG p vrijedi za sva stanja, a AF (AG p) ne vrijedi.





FG p – na svim **pojedinačnim** putovima dolazimo bilo kako do stanja nakon kojega je stalno p=TRUE.

AF (AG p) je striktno jači zahtjev.

AG p vrijedi samo za S_1.

Iz S postoji put natrag na S i nikad ne dođe u S_1.

Zadaci

1. Preslikajte rečenice prirodnog jezika u formule logike LTL:

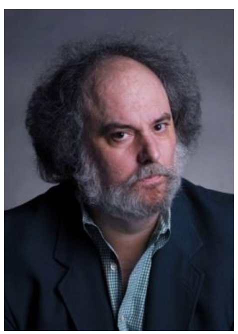
   a) "Nije moguće doći u stanje gdje vrijedi p i ne vrijedi q."

   b) "Uvijek se konačno dolazi u stanje gdje vrijedi p, a od idućeg stanja p vrijedi dalje beskonačno često."

   c) "Uvijek ako vrijedi p, q će vrijediti od tog stanja sve dok p ne prestane vrijediti.“

Potpuna logika vremenskog grananja: CTL*

E. A. Emerson i J. Y. Halpern (1986.)



CTL*

Unificirajuća struktura za CTL i LTL.

Dozvoljeno: 1) Booleove kombinacije F, G, X, U
2) Ugniježđivanje prije primjene E, A

Primjer 1: $A\,[(p\ U\ r) \vee (q\ U\ r)]$

"Duž svih putova p je istinit do r, ili q je istinit do r." Nije CTL, a za LTL samo ako vrijedi za sve pojedinačne putove.

Primjer 2: $E\,(GF\,p)$

"Postoji put na kojem je p istinit beskonačno često." Nije ni CTL ni LTL.

Primjer3: $A\,[X\,p \vee XX\,p]$

"Duž svih putova, p je istinit u sljedećem stanju ili u prvom stanju nakon sljedećeg." Nema ekvivalenta u CTL. U LTL-u samo ako vrijedi za sve pojedinačne putove.

CTL* - definicija sintakse

Formula stanja (dobro formirana CTL* formula):

φ: $p \;|\; \neg p \;|\; p \wedge q \;|\; p \vee q \;|\; p \Rightarrow q \;|\; p \Leftrightarrow q \;|\; E\,\phi \;|\; A\,\phi$

Formula puta:

ϕ: $\varphi \;|\; \neg\phi \;|\; \phi \wedge \phi \;|\; \phi \vee \phi \;|\; \phi \Rightarrow \phi \;|\; \phi \Leftrightarrow \phi \;|\; X\,\phi \;|\; F\,\phi \;|\; G\,\phi \;|\; \phi\, U\, \phi$

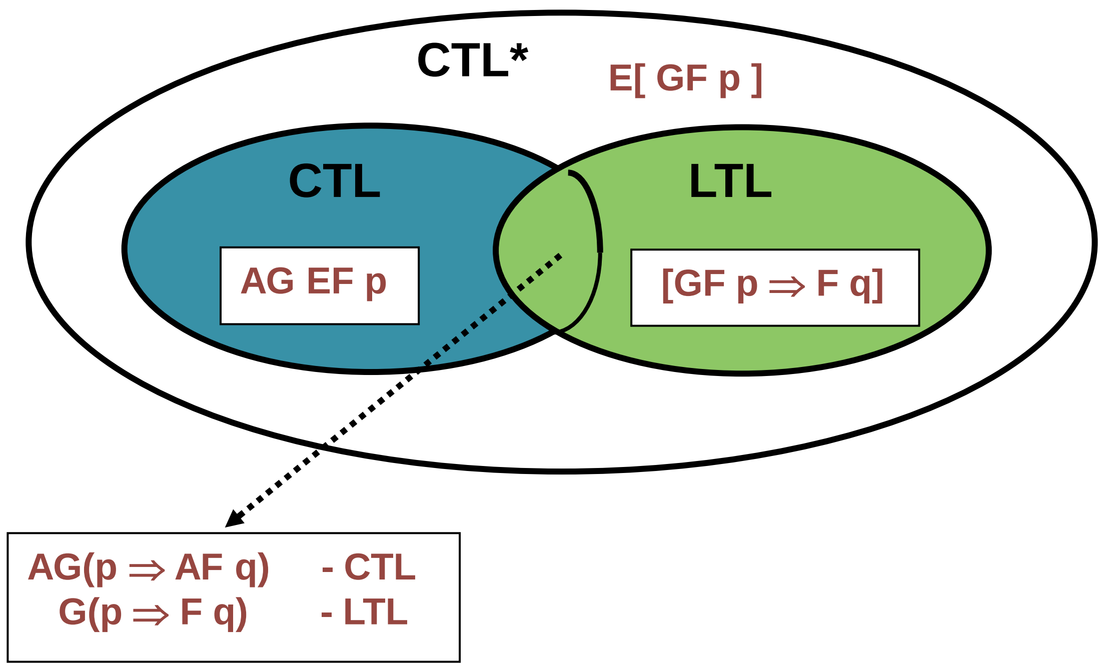
CTL je restriktivni CTL* za formulu puta:

ϕ: Ako su p, q formule stanja, tada su X p, F p, G p, p U q formule puta.

LTL form. stanja φ je ekvivalentna CTL* form. stanja A[φ], dok E[φ] nije dozvoljen, a formula puta je:

ϕ: $p \;|\; \neg\phi \;|\; \phi \wedge \phi \;|\; \phi \vee \phi \;|\; \phi \Rightarrow \phi \;|\; \phi \Leftrightarrow \phi \;|\; X\,\phi \;|\; F\,\phi \;|\; G\,\phi \;|\; \phi\, U\, \phi$

CTL* - odnosi između vrem. logika



Objašnjenja formula sa slike

AG EF p	Iz svakog stanja možemo doći u stanje gdje je p=T. CTL, ali nije u LTL (ranije pokazano).
[GF p $\Rightarrow$ F q]	Ako na svim putovima vrijedi beskonačno često p=T, onda vrijedi i konačno q=T. Npr. beskonačno česti REQ implicira konačni ACK. LTL, ali ne CTL (pravednost se ne može izravno izraziti u CTL-u, a može u LTL-u). Nije isto kao AG AF p $\Rightarrow$ AF q (ako p vrijedi beskonačno često, onda q vrijedi konačno)
E[GF p]	Postoji put s beskonačno često p=T CTL*, ali nije u CTL (dokaz o nemogućnosti izražavanja ovakvog svojstva je složen), nije LTL jer je LTL ekvivalentan CTL* formuli A[φ], a ovdje je egzistencijski operator
AG (p $\Rightarrow$ AF q)	CTL, svaki p=T će konačno slijediti sa q=T
G (p $\Rightarrow$ F q)	LTL, ako vrijedi u strukturi M za svaki put posebno

Složenost provjere modela

Logika	Dokazana složenost provjere modela u odnosu na |φ|*
LTL	PSpace-Complete
CTL	P-Complete
CTL*	PSpace-Complete

Složenost provjere modela u sve tri logike je **linearna** u odnosu na **broj stanja i prijelaza Kripkeove strukture M**.

Problem je što je **broj stanja eksponencijalan** (ili još gori) u ovisnosti o broju varijabli

Poznato je da vrijedi:

$P \subseteq NP \subseteq PSPACE \subseteq EXPTIME \subseteq NEXPTIME \subseteq EXPSPACE \subseteq 2\text{-}EXPTIME \subseteq ELEMENTARY$

* |φ| - broj povezujućih elemenata (npr. za CTL to su logički operatori i operatori stanja AF, EU, EX...) koje čine formulu φ

Eksplicitni postupci računanja skupova stanja koja zadovoljavaju CTL-specifikaciju

Definicija problema

- Za danu Kripkeovu strukturu (usmjereni označeni graf) i određen skup početnih stanja S_0, provjeri da je CTL formula zadovoljena za ta stanja:

Formalno:

- $M, S_0 \models \phi$, tj. $\forall s_0 \in S_0 \quad M, s_0 \models \phi$

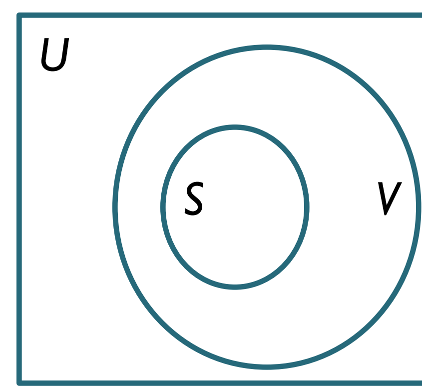
- Uobičajeni pristup: potrebno je pronaći sva stanja koja zadovoljavaju CTL formulu ϕ i ispitati je li željeni podskup S_0 uključen.
- Problem: učinkovit algoritam izračunavanja stanja

Postupci izračunavanja skupova stanja u verifikaciji sustava provjerom modela

- **Eksplicitno** predstavljanje i izračunavanje skupova stanja
  - Kripkeova struktura predstavljena je u memoriji računala kao označeni usmjereni graf (engl. *labeled directed graph*). Izračunavanje skupova stanja koji zadovoljavaju formulu vremenske logike izvodi se postupkom "čvrste točke".
- **Simbolički postupci** predstavljanja i izračunavanja skupova stanja
  - Skupovi stanja i relacija zadane Kripkeove strukture predstavljeni su Booleovim (logičkim) formulama. Booleove formule se u drugom koraku učinkovito predstavljaju binarnim dijagramima odlučivanja (BDD). Izračunavanje skupova stanja koja zadovoljavaju formulu vremenske logike također se izvodi postupkom "čvrste točke".

Notacija

- $v \in V$ Element *v* je član skupa *V*
- $v \notin V$ Element *v* nije član skupa *V*
- $|V|$ Kardinalnost (broj elemenata) skupa V
- $S \subseteq V$ Skup *S* je podskup skupa *V*
- $\varnothing$ Prazan skup (član svih skupova)
- S' Komplement skupa S
- U Svemir – nadskup svih skupova; vrijedi: $S' = U - S$



Notacija

- Partitivni skup skupa V (engl. *power set*):

To je skup skupova $\{S\}$ takvih da je svaki S podskup skupa V. Osim oznake 2^V za sve podskupove skupa V, često se koristi i oznaka **P(*V*).**

$$P(V) = 2^V = \{S \mid S \subseteq V\}$$

$|P(V)| = |2^V| = 2^{|V|}$ Kardinalnost partitivnog skupa od skupa V je potencija broja 2

Primjer: V = {1, 2, 3}

$P(V) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\}$

Relacija kao skup

Binarna relacija **na skupu stanja S:** $\boldsymbol{R \subseteq S \times S}$
je Kartezijev produkt; daje skup uređenih parova elemenata skupa *S*.

Totalna binarna relacija Kripkeove strukture i uređenost:
Za svaki element *s* (u našem kontekstu "stanje sustava") iz skupa *S* postoji barem jedan (može i više) element *t* (u našem kontekstu "stanje sustava") takav da su elementi (stanja) *s* i *t* povezani relacijom *R*:

$\forall s \in S \ \{\exists t \in S \mid (s,t) \in R\}$ Kripke: svaki $s \in S$ je obuhvaćen u *R*.

Skup svih stanja {*t*} čine stanja dosezljiva (engl. *reachable*) u jednom koraku iz skupa stanja {s}.
Totalna binarna relacija *R* na Kripkeovoj strukturi je skup svih mogućih tranzicija (prijelaza) između stanja.

Definicije

$$\forall s \in S \ \{\exists t \in S \mid (s,t) \in R\}$$

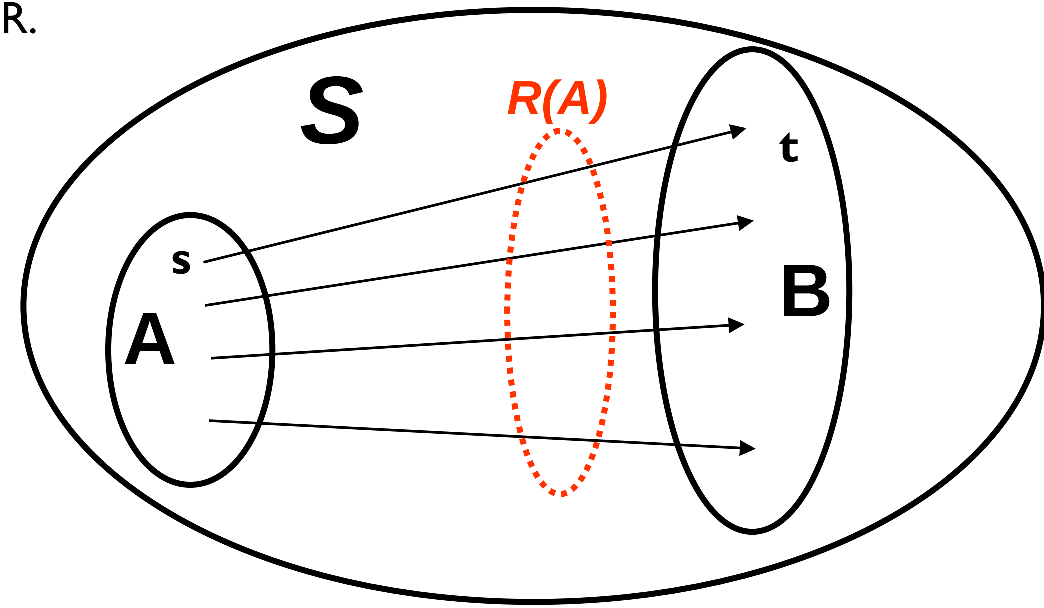
Skup {*t*} je **slika (engl. *IMAGE*)** skupa {*s*} pod relacijom *R*.

Skup {*s*} je **pred-slika (engl. *PRE-IMAGE*)** skupa {*t*} pod relacijom *R*.

Skup {*s*} je **slika (engl. *IMAGE*)** skupa {*t*} pod relacijom R^{-1} (inverznom).

Inverzna relacija

- Neka na skupu S postoji relacija $R \subseteq A \times B$, gdje je: $A = \{s\}$, $B = \{t\}$.
- A = slika (IMAGE) pod **inverznom relacijom** R^{-1}
- B = slika (IMAGE) od A pod R.
- $\mathbf{A = R^{-1}(B)}$
- $\mathbf{B = R(A)}$



- $R(s) = \{t \in S \mid (s,t) \in R\}$
- $R^{-1}(t) = \{s \in S \mid (s,t) \in R\}$
- Primjena inverzne relacija R^{-1} na stanje *t* daje jedno ili više stanja *s* iz kojih u jednom koraku dolazimo do specificiranog stanja *t*.
- $R^{-1}(T) = \cup_{t \in T} R^{-1}(t)$ ako je *T* skup $\{t\}$, rezultat je unija svih $\{s\}$.

Za Kripkeovu strukturu: $R^{-1}(S) = S$ prethodnici svih stanja su sva stanja (*R* je totalna relacija).

$R^{-1}(\varnothing) = \varnothing$ nema prethodnika praznog skupa

Izračunavanje slike

- Postupci izračunavanja slike ili pred-slike preko relacije *R*, ili preko inverzne relacije R^{-1}, predstavljaju najznačajniji dio analize doseziljivih stanja (engl. *reachability analysis*) u sustavima s prijelazima (engl. *transition systems*).
- Neki postupci temelje se na izravnom (eksplicitnom) izračunavanju stanja.
- Neki drugi postupci uvode transformacije preko logičkog kodiranja, pa se slika računa u transformiranom prostoru i zatim dekodira – algoritam za izračun doseziljivih stanja u tom slučaju ćemo pokazati kasnije
- Pretpostavljamo da postoji algoritam izračuna $R(s)$ i $R^{-1}(t)$.

Logičke rekurzije CTL-formula

$AG\ \varphi = \varphi \wedge AX\,AG\ \varphi$; sada i na svim putovima
; počevši od sljedećeg

$EG\ \varphi = \varphi \wedge EX\,EG\ \varphi$; sada i na jednom putu
; počevši od sljedećeg

$AF\ \varphi = \varphi \vee AX\,AF\ \varphi$; sada ili za svako sljedeće
; stanje vrijedi AF φ

$EF\ \varphi = \varphi \vee EX\,EF\ \varphi$; sada ili za jedno sljedeće
; stanje vrijedi EF φ

$A[\varphi\ U\ \psi] = \psi \vee (\varphi \wedge AX\,A(\varphi\ U\ \psi))$
; ψ vrijedi sada ili
; φ vrijedi sada i za svako
; sljedeće stanje vrijedi
; $A(\varphi\ U\ \psi)$

$E[\varphi\ U\ \psi] = \psi \vee (\varphi \wedge EX\,E(\varphi\ U\ \psi))$; slično kao AU, ali
; samo za jedan put

Izračunavanje EX, EG, EU (adekvatan skup) omogućuje izračunavanje svih CTL-formula.

Zamjena logičkih operacija operacijama nad skupovima

Model M = (S, R, L) EX, EG, EU - adekvatan skup

$R(s) = \{ t \in S \mid (s, t) \in R \}$ - daje sljedbenike stanja *s* (skup *t*-ova)

$Q(\text{False}) = \varnothing$
$Q(\text{True}) = S$
$Q(p) = \{ s \mid p \in L(s) \}$ - skup stanja s u kojima vrijedi p = True
$Q(\neg f) = S - Q(f)$ - sva stanja u S osim onih u kojima f = True
$Q(f \wedge g) = Q(f) \cap Q(g)$ - skup dobiven presjekom skupova

$Q(EX\ f) = \{ s \mid R(s) \cap Q(f) \}$ - stanja koja imaju sljedbenike u Q(f) (R(s) daje sve sljedbenike u jednom koraku)

$Q(EG\ f) = Q(f) \cap Q(EX\ EG\ f)$ - stanja Q(f) u kojima je f = True i stanja za koja vrijedi Q(EG f) nakon jednog koraka (EX)

$Q[E(f\ U\ g)] = Q(g) \cup [Q(f) \cap Q(EX\ E(f\ U\ g))]$ - stanja u kojima je g = True, ili stanja u kojima je f = True i nakon jednog koraka (EX) vrijedi E(f U g)

Izračunavanje skupa stanja za CTL-formulu EX

Zadan je skup stanja Q(*f*) u kojima je istinita formula vremenske logike *f*. Potrebno je pronaći skup stanja **Q(EX *f*)**, dakle ona stanja iz kojih u jednom koraku dolazimo do nekog stanja iz Q(*f*).

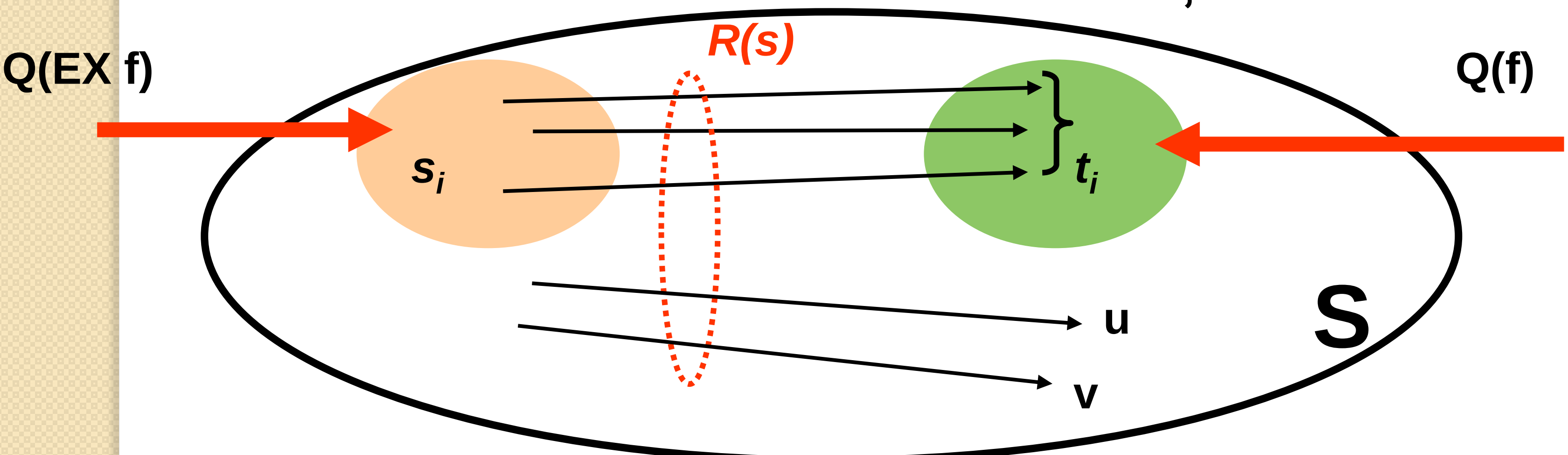
$$Q(\, EX\, f\,) = \{\, s \mid R\,(\, s\,) \cap Q(\, f\,) \neq \emptyset\,\}$$

$$= \{\, s \mid \exists_{t \,\in\, R\,(\,s\,)}\, t \in Q(\, f\,)\}$$

$$= \{\, s \mid \exists_{t \,\in\, Q\,(\,f\,)}\, (s, t) \in R\}$$

$$\boxed{= R^{-1}\,(\, Q\,(\, f\,))}$$

Slika pod inverznom relacijom



CTL-operatori kao skupovi stanja

1. Izračunavanje skupa **Q(EX f)** (pokazano ranije):

$$Q(EX\ f) = R^{-1}\ (Q(f))$$

2. Izračunavanje skupa **Q(EG f)** je uz supstituciju za EX:

$$Q(\ EG\ f\ ) = Q(f\ ) \cap Q(\ EX\ EG\ f\ )$$

$$Q(\ EG\ f\ ) = Q(f\ ) \cap R^{-1}\ (Q(\ EG\ f\ ))$$

3. Izračunavanje skupa **Q(E (f U g))** je uz supstituciju za EX:

$$Q[\ E(\ f\ U\ g)] = Q(\ g\ ) \cup [Q(\ f\ ) \cap Q(\ EX\ E(\ f\ U\ g))]$$

$$Q[\ E(\ f\ U\ g)] = Q(\ g\ ) \cup [Q(\ f\ ) \cap R^{-1}\ (Q(\ E(\ f\ U\ g)))]$$

Za izračunavanje 2. i 3. potrebna je teorija "čvrste točke" jer nije očigledno kako razriješiti rekurzije.

Fiksna (čvrsta) točka (1/6)

- Monotone funkcije i fiksna točka

Definicije:

S	- skup stanja
F: P(S) $\rightarrow$ P(S)	- funkcija na svim podskupovima u S,
P(S)	- 2^S (partitivni skup)

1. F je monotona akko
   X $\subseteq$ Y implicira (povlači) F(X) $\subseteq$ F(Y)
   za sve podskupove X i Y u S (oznaka $\subseteq$ - podskup).

2. Podskup X od skupa S je fiksna točka (engl. *fix-point*) funkcije F akko:

$$F(X)=X$$

Fiksna (čvrsta) točka (2/6)

Primjer 1:
Neka je $S = \{s_0, s_1\}$,
te neka je $F(Y) = Y \cup \{s_0\}$ za sve podskupove $Y \subseteq S$.

Test na monotonost:
Neka je Y' također bilo koji podskup od S.
Svaki $Y' \subseteq Y$, implicira $Y' \cup \{s_0\} \subseteq Y \cup \{s_0\}$, te je F monotona.

Analiza fiksne točke (za sve podskupove $Y \subseteq S=\{s_0, s_1\}$):
Podskup { } nije fiksna točka jer $F(\{\ \}) = \{\ \} \cup \{s_0\} = \{s_0\}$.
Podskup $\{s_0\}$ je najmanji *fix-point*, jer $F(\{s_0\})=\{s_0\} \cup \{s_0\}=\{s_0\}$.
Podskup $\{s_1\}$ nije fiksna točka jer $F(\{s_1\}) = \{s_1\} \cup \{s_0\} = \{s_0, s_1\}$.
Skup $\{s_0, s_1\}$ je najveći *fix-point*, jer $F(\{s_0, s_1\})=\{s_0, s_1\}\cup\{s_0\}=\{s_0, s_1\}$.

Monotone funkcije uvijek imaju najmanju i najveću fiksnu točku.

Funkcije za izračunavanje skupova stanja u Kripkeovoj strukturi koje nas zanimaju su monotone te imaju najmanji i najveći *fix-point*:

$Q(\,EG\ f\,) = Q(f) \cap Q(\,EX\ EG\ f\,)$
$Q(\,E(f\ U\ g)\,) = Q(g) \cup [Q(f) \cap Q(\,EX\ E(\,f\ U\ g\,))]$

Fiksna (čvrsta) točka (3/6)

Primjer 2:
$S = \{s_0, s_1\}$
Funkcija: $G(Y)$ = ako $[Y = \{s_0\}]$ tada $\{s_1\}$ inače $\{s_0\}$

Test na monotonost:
Primjena funkcije G na $Y = \{s_0 , s_1\}$ daje $\{s_0\}$.
Primjena funkcije G na $Y' = \{s_0\}$ daje $\{s_1\}$.
Y' je podskup od Y , tj. $(Y' \subseteq Y)$,
ali kako rezultat $\{s_1\}$ nije podskup od $\{s_0\}$ to G nije monotona.

Analiza fiksne točke (za sve podskupove $Y \subseteq S=\{s_0, s_1\}$):
$G(\{\}) = \{s_0 \}$
$G(\{s_0\}) = \{s_1\}$
$G(\{s_1 \}) = \{s_0 \}$
$G(\{s_0, s_1\} = \{s_0\}$

G(Y) nema nijednu fiksnu točku. **Nemonotone funkcije mogu, ali i ne moraju imati fiksnu točku.**

Fiksna (čvrsta) točka (4/6)

Postupak izračunavanja fiksne točke: Teorem Knaster-Tarski

Neka je S skup: $S = \{s_0, s_1, \ldots, s_n\}$ sa n+1 elementom.

Označimo sa F^i : funkcija F primijenjena i-puta, odnosno: $F(F(\ldots F(X)))$

Npr. Neka je $F(Y)= F^1(Y)=Y \cup \{s_0\}$ gdje je $Y \subseteq S$
$F^2(Y) = F(F(Y)) = [Y \cup \{s_0\}] \cup \{s_0\} = Y \cup \{s_0\} = F(Y)$, te je $F^2 = F$
Za ovaj primjer vrijedi: $F^i = F$ za sve $i \geq 1$

Teorem [P(S) je partitivni skup]:

Ako je $F: P(S) \rightarrow P(S)$ **monotona**, tada

$F^{n+1}(\emptyset)$ je najmanji *fix-point* od F.
$F^{n+1}(S)$ je najveći *fix-point* od F.

Fiksna (čvrsta) točka (5/6)

Dokaz da je $F^{n+1}(\emptyset)$ najmanji *fix-point* od F: (napomena: dokaz nije potrebno učiti)

F je monotona (uvjet) pa vrijedi $\emptyset \subseteq F(\emptyset)$, također $F(\emptyset) \subseteq F(F(\emptyset))$, odnosno $F^1(\emptyset) \subseteq F^2(\emptyset)$.
Indukcijom slijedi:
$F^1(\emptyset) \subseteq F^2(\emptyset) \subseteq \ldots \subseteq F^i(\emptyset)$ za sve $i \geq 1$

Definiramo: **i = n + 1** (**n + 1** = broj elemenata u skupu S).
Tvrdimo: jedan od gornjih $F^k(\emptyset)$ je *fix-point*, tj. $F(F^k(\emptyset))=F^k(\emptyset)$

Kad $F^1(\emptyset)$ ne bi bio *fix-point* onda bi $F^1(\emptyset)$ morao sadržavati najmanje 1 element više od $\emptyset$ (jer tada $\emptyset \neq F(\emptyset)$).
$F^2(\emptyset)$ bi morao sadržavati barem 2 elementa, morao biti veći od $F^1(\emptyset)$.
Svaki daljnji bi morao imati barem jedan element više od prethodnika.

Kad $F^{n+1}(\emptyset)$ ne bi bio *fix-point*, $F^{n+2}(\emptyset) = F(F^{n+1}(\emptyset))$ bi morao imati n+1+1 element, što je nemoguće jer S ima samo n+1 elemenata. Dakle $F^{n+1}(\emptyset)$ mora biti *fix-point*.

Odnosno: $F(F^{n+1}(\emptyset)) = F^{n+1}(\emptyset)$ $F^{n+1}(\emptyset)$ je fiksna točka

Fiksna (čvrsta) točka (6/6)

Još treba dokazati da je to najmanja fiksna točka!

Neka je X neki drugi *fix-point* od F, tj. F(X) = X
Moramo pokazati da je $F^{n+1}(\emptyset) \subseteq X$.

Kako je $\emptyset \subseteq X$, to slijedi $F(\emptyset) \subseteq F(X) = X$ (jer je funkcija monotona)
Dakle: $F(\emptyset) \subseteq X$.
$F^2(\emptyset) \subseteq F(F(X)) = X$ (jer je X fix-point)
Indukcijom $F^i(\emptyset) \subseteq X$ za sve $i \geq 0$, pa i za $i = n + 1$, slijedi $F^{n+1}(\emptyset) \subseteq X$

Dokaz za najveći fix-point analogno uz zamjenu: $\subseteq$ sa $\supseteq$, te $\emptyset$ sa S.

Teorem daje ujedno i algoritam izračunavanja i garantira završetak:

<u>Najmanji *fix-point*</u>: iterativna primjena F na prazan skup $\emptyset$, dok rezultat ne postane invarijantan na tu primjenu.

<u>Najveći fix-point</u>: iterativna primjena F na skup svih stanja S, dok rezultat ne postane invarijantan na tu primjenu.

Najveća gornja granica broja iteracija: n+1 (za S sa n+1 elementom)

Izračunavanje EG preko najveće fiksne točke (1/3)

- $Q(\ EG\ f\ ) = Q(\ f\ ) \cap Q(\ EX\ EG\ f\ )$
  $= Q(\ f\ ) \cap R^{-1}(Q(EG\ f))$

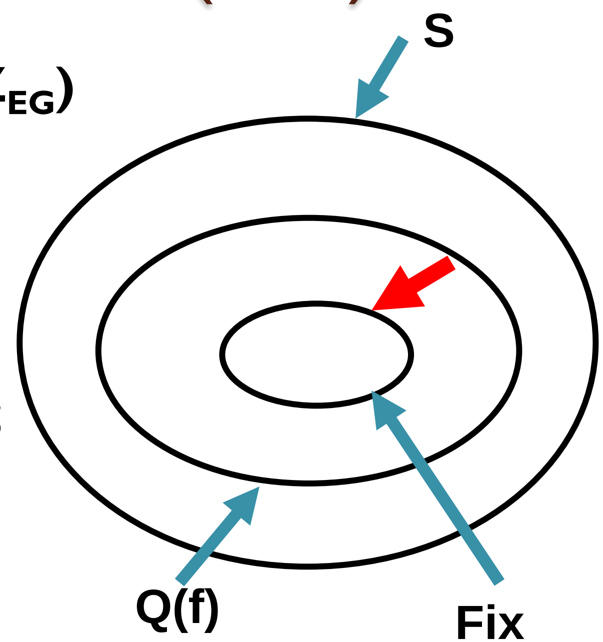
- Primjenom funkcije $F(X) = Q(\ f\ ) \cap R^{-1}(X)$ i to n+1 puta na skup svih stanja S slijedi najveća čvrsta točka, tj. podskup Z_{EG} koji zadovoljava CTL-formulu EG f:

  $F^{n+1}(S) = Q(EG(f))$

- $Z_{EG} = F_{EG}\ (Z_{EG}) = \underbrace{Q(\ f\ ) \cap R^{-1}}_{\mathbf{F_{EG}}}(Z_{EG})$

Izračunavanje EG preko najveće fiksne točke (2/3)

Q(EG f): $Z_{EG} = F_{EG}(Z_{EG}) = Q_f \cap R^{-1}(Z_{EG})$



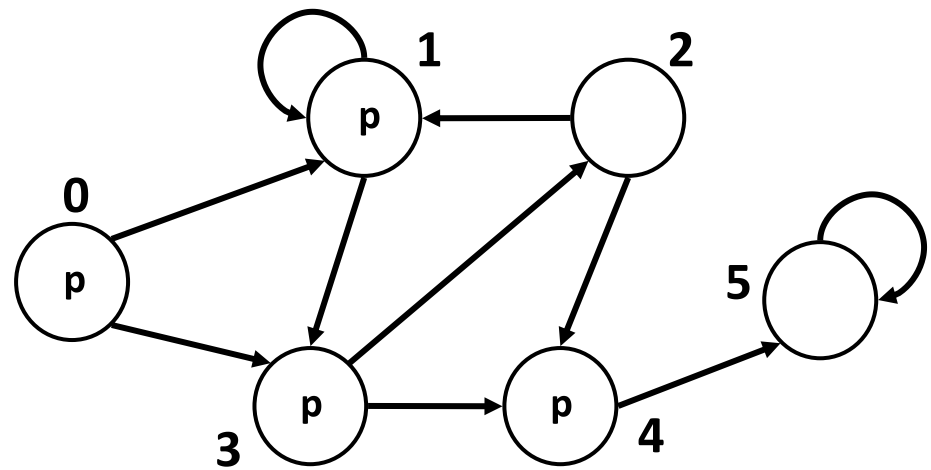
Q(EG f) (Q(f)): {
k := 0; Z_k := S;
do {
$Z_{k+1} := Q(f) \cap R^{-1}(Z_k)$;
if ($Z_{k+1} = Z_k$) return Z_k ;
k++ ;
} forever;
}

Započinjemo sa skupom S, tj. $Z_0 = S$, i prva iteracija daje:
$Z_1 = Q(f) \cap R^{-1}(S) = Q(f)$, dakle u prvoj iteraciji je $Z_1 \neq Z_0$ te se ide dalje:
$Z_2 = Q(f) \cap R^{-1}(Q(f))$ … itd. sve dok $Z_{n+1} = Z_n$ tj. dosegne fiksnu točku

Budući da $R^{-1}(S) = S$, bolje je odmah započeti sa $Z_k = Q(f)$.

Izračunavanje EG preko najveće fiksne točke (3/3)

<u>Za primjer sa slike odredi stanja za koja vrijedi EG p:</u>



Q(p) = { 0, 1, 3, 4 }, tu je p=True

$$\boxed{Z_{k+1} = Q(p) \cap R^{-1}(Z_k)}$$

Početno: $Z_0 = S = \{ 0, 1, 2, 3, 4, 5 \}$

- $R^{-1}(Z_0) = R^{-1}(S) = S$ $= \{ 0, 1, 2, 3, 4, 5 \}$
- $Z_1 = \{ 0, 1, 3, 4 \} \cap R^{-1}(\{ 0, 1, 2, 3, 4, 5 \})$ $= \{ 0, 1, 3, 4 \}$ $\neq Z_0$
- $R^{-1}(Z_1) = R^{-1}(\{ 0, 1, 3, 4 \})$ = prethodnici $= \{ 0, 1, 2, 3 \}$
- $Z_2 = \{ 0, 1, 3, 4 \} \cap R^{-1}(Z_1)$ $= \{ 0, 1, 3 \}$ $\neq Z_1$
- $R^{-1}(Z_2) = R^{-1}(\{ 0, 1, 3 \})$ = prethodnici $= \{ 0, 1, 2 \}$
- $Z_3 = \{ 0, 1, 3, 4 \} \cap R^{-1}(Z_2)$ $= \{ 0, 1 \}$ $\neq Z_2$
- $R^{-1}(Z_3) = R^{-1}(\{ 0, 1 \})$ = prethodnici $= \{ 0, 1, 2 \}$
- $Z_4 = \{ 0, 1, 3, 4 \} \cap R^{-1}(Z_3)$ $= \{ 0, 1 \}$ $= Z_3$ (fiksna točka)

Rješenje: stanja {0, 1} zadovoljavaju EG p.

Izračunavanje EU putem najmanje fiksne točke (1/2)

$Q[\, f\ EU\ g] = Q(\, g\,) \cup [Q(\, f\,) \cap Q(\, EX\ E(\, f\ U\ g))]$

$Q[\, E(\, f\ U\ g)] = Q(\, g\,) \cup [Q(\, f\,) \cap R^{-1}\ (Q(\, E(\, f\ U\ g))]$

Primjenom funkcije $F(X) = Q(\, g\,) \cup [Q(\, f\,) \cap R^{-1}(X)]$ i to n+1 puta na prazan skup $\varnothing$ slijedi najmanja čvrsta točka, tj. podskup Z_{EU} koji zadovoljava CTL formulu E(f U g):

$F^{n+1}(\varnothing) = Q(E\ (f\ U\ g))$

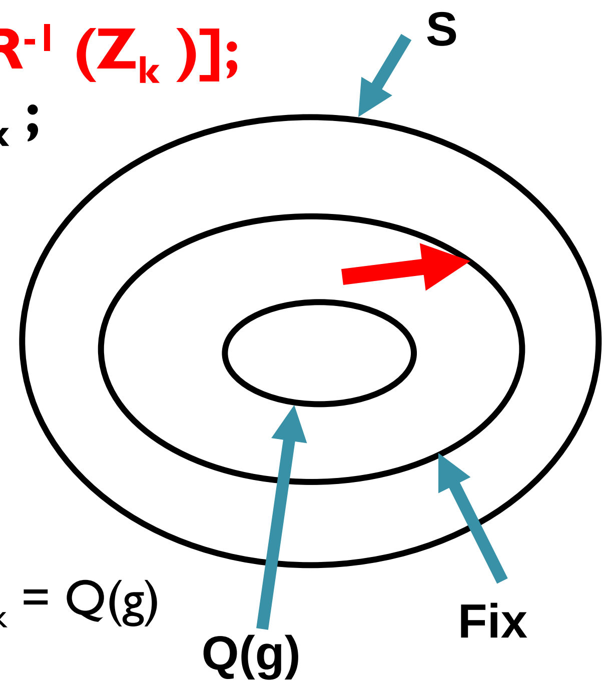
- $Z_{EU} = F_{EU}(Z_{EU}) = Q(\, g\,) \cup Q(\, f\,) \cap R^{-1}(Z_{EU})$

$\mathbf{F_{EU}}$

Izračunavanje EU putem najmanje fiksne točke (2/2)

Q(f EU g): $Z_{EU} = F_{EU}(Z_{EG}) \;=\; Q_g \cup [Q_f \cap R^{-1}(Z_{EU})]$

```
Q(f EU g) (Q(f), Q(g))        {
        k := 0;   Zk := ∅;
        do      {
                Zk+1 := Q(g) ∪ [Q(f) ∩ R-1 (Zk )];
                if (Zk+1 = Zk)   return Zk ;
                k++ ;
        } forever;
}
```

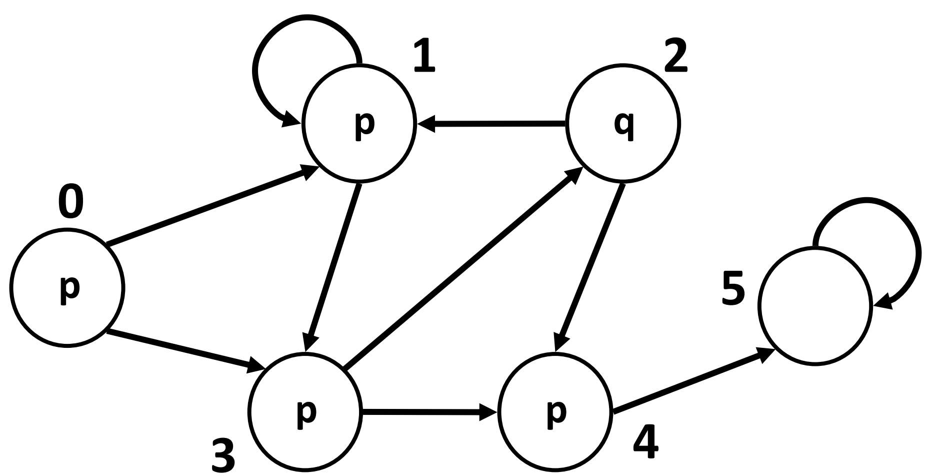


Započinjemo s praznim skupom $Z_0 = \emptyset$.

Budući da $R^{-1}(\emptyset) = \emptyset$, bolje odmah započeti sa $Z_k = Q(g)$

1. zadatak

- Za primjer sa slike odredite stanja za koja vrijedi E(p U q) korištenjem algoritma za izračunavanje EU pomoću najmanje fiksne točke.



2. zadatak

- Za funkciju $F(X) = (X \cup \{s1\}) \cap \{s2\}$ i skup mogućih stanja $S = \{s0, s1, s2\}$ odredite:

a) Je li funkcija monotona.

b) Ako funkcija jest monotona, nađite najmanju i najveću fiksnu točku.

Zadatak za bonus bod

- Koristeći teoriju fiksne točke i odgovarajući algoritam, odredite Q(EG r) za Kripkeovu strukturu prikazanu na slici. Potrebno je napisati cjelokupni postupak dobivanja rješenja i konačno rješenje.